

# EXERCICE STATION MÉTÉO

---

## VERSION LONGUE

### Table des matières

1	Projet station météo .....	2
1.1	Objectif .....	2
1.2	Les attendus .....	2
1.2.1	Qualité .....	2
1.2.2	Cahier des charges.....	2
2	Déroulé de l'exercice.....	3
2.1	Maquette.....	3
2.2	Intégration.....	3
2.3	Dynamisme.....	3
2.3.1	Date et heure.....	3
2.3.2	Météo et éphéméride .....	3
2.3.3	Les saints du jour .....	3
3	Annexes .....	4
3.1	Outils .....	4
3.2	APIs.....	4
3.2.1	API météo et éphéméride .....	4
3.2.2	API saint du jour .....	4

# 1 Projet station météo

## 1.1 Objectif

Le but de cette application est de créer un site météo contenant l'éphéméride du jour ainsi que les températures et la météo prévisionnelle à 2 jours. Dans un premier temps, ce sera la météo de la ville de Toulouse qui sera affichée.

## 1.2 Les attendus

### 1.2.1 Qualité

Le code doit passer la validation, que ce soit CSS ou HTML.

Les pages doivent être adaptables sur desktop, tablettes et mobiles, avec une réflexion *desktop first*.

Le code backend doit :

- Respecter les critères de sécurité
- Être facilement maintenable
- Être commenté
- Être documenté

### 1.2.2 Cahier des charges

La page d'accueil du site affichera pour la position actuelle de l'utilisateur (ou par défaut pour Toulouse) :

- La date actuelle (format au choix mais français)
- L'heure actuelle à la seconde près
- L'heure de lever et de coucher du soleil
- Les saints du jour, ou au moins le saint principal s'il y en a plusieurs
- La météo prévisionnelle du jour et des deux jours suivants
- Les températures prévisionnelles du jour
- Un lien vers un formulaire de contact

Un formulaire de contact sera également disponible proposant des contrôles permettant à l'utilisateur de fournir :

- Son nom
- Son adresse de contact (téléphone, courriel, twitter ou autre réseaux sociaux)
- Son message

## 2 Déroulé de l'exercice

### 2.1 Maquette

Créez la maquette du projet, qui doit présenter chaque page telle que vous l'imaginez, au moins en version desktop et en version mobile.

### 2.2 Intégration

Intégrez la maquette en HTML et faites la mise en forme avec CSS.

Pensez aux versions adaptées aux différentes taille d'écran.

### 2.3 Dynamisme

#### 2.3.1 Date et heure

Automatisez l'affichage de la date et de l'heure avec une tâche de fond appelée toutes les secondes.

Le changement de jour provoquera la mise à jour de l'éphéméride.

#### 2.3.2 Météo et éphéméride

Le site suisse <https://prevision-meteo.ch/> fournit une API permettant de récupérer la météo du jour et des jours suivants, ainsi que l'éphéméride à l'exception des saints du jour. Utilisez-là pour récupérer et injecter les données dans votre page.

#### 2.3.3 Les saints du jour

Le site <https://nominis.ccf.fr/json/nominis.php> propose une API donnant les saints du jour. Cependant cette API réclame l'utilisation d'un module à installer et activer sur votre navigateur : voyez en annexe les informations à ce sujet.

## 3 Annexes

### 3.1 Outils

Le validateur HTML du W3C : [https://validator.w3.org/#validate\\_by\\_input](https://validator.w3.org/#validate_by_input)

Le validateur CSS du W3C : [https://jigsaw.w3.org/css-validator/#validate\\_by\\_input](https://jigsaw.w3.org/css-validator/#validate_by_input)

Un vérificateur WCAG en ligne : <https://achecker.achecks.ca/checker/index.php>

### 3.2 APIs

#### 3.2.1 API météo et éphéméride

<https://prevision-meteo.ch/>

Pour obtenir tout le nécessaire, il suffit d'interroger <https://www.prevision-meteo.ch/services/json/toulouse>. L'avantage avec cette API c'est qu'elle vous fournit le texte descriptif de la météo, et même les icônes correspondantes au temps prévisionnel. En revanche, il y a beaucoup d'information et le JSON retourné contient beaucoup de choses qui ne nous intéressent pas, puisqu'on a le temps heure par heure pendant 7 jours. Il faudra donc bien analyser le contenu pour en extraire ce qui nous intéresse. Mais tout y est. Il est aussi possible que cette API ne fournisse pas le temps pour toutes les localités françaises, mais Toulouse y est...

La documentation se trouve sur <https://www.prevision-meteo.ch/uploads/pdf/recuperation-donnees-meteo.pdf>.

#### 3.2.2 API saint du jour

Pour pouvoir utiliser cette API, il faut installer un module sur votre navigateur (voir [l'énoncé de l'exercice](#))

<https://nominis.cef.fr/json/nominis.php>

Cette API vous retourne un gros JSON contenant tous les saints du jour et leur descriptions ainsi que leur vie... Là encore, il faudra faire un tri !

Notez que pour utiliser cette API il est nécessaire d'ajouter une extension à votre navigateur pour outrepasser les contrôles *CORS*. CORS (Cross Origin Resource Sharing) empêche les requêtes fetch d'aboutir si elles ne sont pas envoyées sur le même serveur que la page courante, sauf si le serveur a été configuré pour accepter des requêtes de tout le monde. Il faut donc leurrer le navigateur en réécrivant partiellement l'entête retournée par le serveur en lui faisant croire que tout va bien au niveau CORS. Ceci est fait par les extensions suivantes (par exemple, il en existe d'autres) :

- CORS Everywhere pour Firefox
- Moesif Origin & CORS Changer maintenant sur Firefox et Chrome

Évitez Allow CORS qui ne fonctionne pas trop bien sur Windows et pas du tout sur Linux.

Je vous engage à utiliser MOESIF qui fonctionne très bien sur tous les navigateurs que je connais. Notez cependant qu'il faut absolument le désactiver si vous voulez aller sur des serveurs bien configurés, comme votre banque ou google, sinon rien ne fonctionnera correctement. Réservez-le donc à l'usage exclusif de votre page locale.

Si vous voulez en savoir plus sur CORS, lisez <https://developer.mozilla.org/fr/docs/Web/HTTP/CORS>.